Manay Patel - S2AND PIPELINE README
Fall 2021

Required Files by S2AND:

1) patent_papers.json fields:

-"paper_id" (required)
- "title" (required)
- "authors" (required)
- "years" (required)
- "abstract" (optional)
- "journal_name" (optional)
- "references" (required but can be left empty)
- "venue" (optional)

2) patent_signatures.json:

Json file of key-value pairs where the key is signature id and value is an object that represents the signature.
Signature object fields:

- "author_id" [ground truth]
- "paper_id" [paper_id of the signature]
- "signature_id" [each signature has unique id]
- "author_info" : [object of author information]
    - "given block" [block as first letter of initial + last]
    - "block" [both of these fields had same values]
    - "postion" [author credit position in paper]
    - "first"
    - "middle"
    - "last"
    - "suffix"
    - "affiliations" [usualy institute name in scholarly papers but no similar info in patents]
    - "email"

3) patent_cluster.json:

Json file where the key is cluster id and value is cluster object with cluster information. Each cluster object corresponds to a ground truth label, that is, an author. This means that every signature under this cluster will have the same "author_id" field.
Cluster object fields:

- "cluster_id" [conventionally: initial of dataset name + unique id, eg, PV_131 for PatentsView]
- "signature_ids" [array of all signatures that have same "author_id" field]
- "model_version" [unknown field, set it to -1 as seen in other S2AND data]

Pipeline - Patentsview to S2AND

There are some files that contain "CHANGE HERE" in their documentation. These are the files that read different ground truth labels for the test set or the train set. The lines next to "CHANGE HERE" can be easily changed to allow any other files of labelings with similar formatting.

1) new_uid_map_converter.py: Creates a mapping from patentsview paper IDs to new unique IDs. This has to be done because S2AND only accepts integers as paper IDs, but patentsview paper IDs are strings that contain letters.

2) papers_converter.py: Creates err_patent_papers.json which contains papers and their information in S2AND's format. This is an err_ file because it may contain paper objects that are missing some fields which may or may not be required for a paper object to have.

3) paper_err_corrector.py: Reads in err_patent_papers.json and creates a finalized patent_paper.json file that contains only papers that have every required field and any optional fields are left as null or empty. In addition, this script also creates key_list.json this is a list of all papers that have all the required fields so as to not cause any error while creating training/testing files or even while running S2AND.

4) trainPapersSet.py & testPapersSet.py (*Contains CHANG HERE***):** both of these scripts create train_canopies.json and test_canopies.json respectively. These are needed just to faster the converting process for the signatures and clusters files. These canopies are also used to decrease the length of patent_paper.json to decrease runtime for processing data by S2AND.

(optional) reducePapersToCanopies.py: reduces patent_papers.json from all papers to only papers that are needed for training and testing. It creates new_patent_papers.json file, so, if used, remove/change the current "patent_papers.json" and change the name of "new_patent_papers.json" to "patent_papers.json"

5) sigANDcluster_converter.py (*Contains CHANG HERE***):** reads through previously created files, patentsview files, and the train/testing files to create patent_signatures.json and patent_clusters.json.

****End of pipeline for testing files****

Additional steps for training files: The following steps are to create specter embeddings files that is used by S2AND for help with training. Specter embeddings is another repo from AllenAi that creates embeddings for training papers using the paper's abstract.

6) paper_for_specter_training.py: reads papers from patent_signature.json file and creates a train_paper.json which is an input to Specter model that gives us a Specter embeddings file, "specter.json", in return.

6.5) Download the allenai/specter repo and run the following command: (https://github.com/allenai/specter)

*python scripts/embed_papers_hf.py --data-path ***train_paper.json path*** --output ***desired path for specter.json output*** --batch-size 8*

For example: python scripts/embed_papers_hf.py --data-path /iesl/canvas/mbpatel/patent_data/train_paper.json --output /iesl/canvas/mbpatel/patent_data/specter.json --batch-size 8

7) training_to_specter_pickle.py: takes in the specter.json file and converts it into a pickle file, patent_specter.pickle, which is what the S2AND model takes in as input for training datasets.

Directory Setup for ease of PipeLine usage

- Create a main directory (for eg, patent_data) containing “patent.tsv”, “uspatentcitation.tsv”, “rawinventor.tsv”, “rawassignee.tsv” files from patentsview dataset. In addition to this place all of the pipeline scripts in this directory.

- Create a train folder and a test folder containing all the ground truth labels for respective files.

- After obtaining signature and clusters for a train/test file (if train file then also obtain the specter pickle file before this step) move the “patent_signature.json”, “patent_clusters.json” and, if applicable (for training files), then the “patent_specter.pickle” as well to the corresponding train/test folder.
For example, for eval_common_characteristics.train create a folder (say common_charac/) and move the signatures, clusters, and specter pickle file into common_charac/ directory.